LOSSLESS COMPRESSION OF MEDICAL IMAGE BY HIERARCHICAL SORTING

Atsushi Myojoyama, Tsuyoshi Yamamoto

Graduate School of Engineering, Hokkaido University, Sapporo-shi, Hokkaido, Japan

Abstract-**We propose a new lossless compression method for medical images, based on a hierarchical sorting. Hierarchical sorting is a technique that achieves a high compression ratio by detecting the regions where image patterns change abruptly, and by sorting pixel order by value to increase predictability. This method enables control of sorting accuracy along with size and complexity. As a result, we can reduce the sizes of the permutation tables and reuse the tables for other image regions. Comparison of this method through experiment reveals better performance for medical images generated by X-ray CT, MRI and large size CR·DR instruments. This technique applies a quad-tree division method to divide an image into blocks in order to support progressive decoding and fast preview of large images.**
***Keywords* - lossless compression, medical image, sorting algorithm, hierarchical coding**

I. Introduction

In medical imaging, data compression techniques are needed for archiving the many images generated by medical imaging instruments, because medical law mandates that these images be kept for long periods. In keeping with the nature of medical diagnosis, the original image quality must be preserved through compression-decompression. Progressive retrieval is another requirement in this field. It promotes fast diagnosis via rapid data transmission and a quick preview function. When developing medical image compression, we must take into account the image format used in the field. In general, medical images use more bits per pixel than standard photographs. Most medical images use 10- or 12-bit/pixel formats. There are two other compressible aspects of medical images. One is that there are extremely large image formats such as for digital radiography (DR) and computed radiography (CR). These images are represented in 1024×1024-pixel or higher resolutions. The other is the case where a set of many images with similar imaging parameters is generated. Advanced medical imaging instruments such as helical-mode X-ray CT and 3D MRI generate a series of 2D images whose members are very similar to each other. Considering these characteristics, we developed a new compression algorithm that achieves high compression performance for medical images.

In general, image compression algorithms consist of three steps: prediction, modeling and encoding. Prediction is based on the experiential principle that the entropy of the prediction error is smaller than original entropy if the next pixel can be predicted fairly accurately from the already coded data [1]. This principle is fundamental to image compression.

In lossless image compression, the algorithms used for JPEG (lossless mode), an ISO standard, are well known [2]. These algorithms use several prediction methods and entropy-coding methods depending on the image type. For entropy-coding schemes, either Huffman coding or arithmetic coding is used. However, since these compression methods are intended for common photographic images of 8-bit/pixel format, these are not applicable for medical images in 10- to 12-bit/pixel format.

In 1996, Weinberger et al. developed LOCO-I (applied to JPEG-LS) which supports not only 8-bit/pixel but also 16-bit/pixel formats [3]. LOCO-I is a lossless and near lossless compression algorithm which combines the simplicity of Huffman coding with the compression potential of simple fixed context models. Since the method employs a one-pass scheme, the compression speed is usually higher than that of two-pass schemes. The compression ratio of LOCO-I is better than schemes based on arithmetic coding. However, LOCO-I doesn't support progressive reconstruction and it handles only a single image for each compression process.

This paper proposes a new lossless coding method for medical images. To approach the problem with conventional methods, we developed the new technique, *"hierarchical sorting."* This method can achieve a high compression ratio by detecting patterns observed in an image set collected by X-ray CT and MRI instruments. This technique also supports progressive decoding.

II. Theory

If pixels are sorted perfectly in density order, a predictive coder generates the smallest possible predicting code. However, a table is required to restore the pixel positions to recover the original image. The size of the table is usually greater than the size of the original image. Hierarchical sorting does not sort the pixels in an image accurately. This technique generates *"permutation tables"* and applies those tables to many regions of many hierarchical layers. Hierarchical sorting consists of two processes: block division and permutation table creation. In the first process, the original image is divided into sub-blocks. Then, the permutation table is created to sort the pixels in the block, and the block is tested for whether to sort at this level.

Some lossless compression algorithms developed recently employ hierarchical image segmentation for progressive reconstruction. Such algorithms help us quickly understand the detailed characteristics of the image. There are two hierarchical image segmentation methods. One is multi-scale

Report Documentation Page

Report Date	Report Type	Dates Covered (from... to)
25 Oct 2001	N/A	-

Title and Subtitle Lossless Compression of Medical Image by Hierarchical Sorting	**Contract Number**
	Grant Number
	Program Element Number
Author(s)	**Project Number**
	Task Number
	Work Unit Number
Performing Organization Name(s) and Address(es) Graduate School of Engineering Hokkaido University Sapporo-shi Hokkaido, Japan	**Performing Organization Report Number**
Sponsoring/Monitoring Agency Name(s) and Address(es) US Army Research, Development & Standardization Group (UK) PSC 802 Box 15 FPO AE 09499-1500	**Sponsor/Monitor's Acronym(s)**
	Sponsor/Monitor's Report Number(s)
Distribution/Availability Statement Approved for public release, distribution unlimited	
Supplementary Notes Papers from 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, October 25-26, 2001 held in Istanbul, Turkey. See also ADM001351 for entire conference on cd-rom., The original document contains color images.	
Abstract	
Subject Terms	
Report Classification unclassified	**Classification of this page** unclassified
Classification of Abstract unclassified	**Limitation of Abstract** UU
Number of Pages 4	

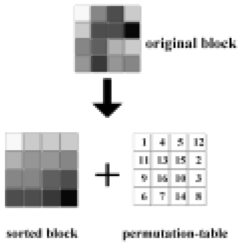

Fig. 1. Relations among original block, permutation table and sorted block

encoding of an image based on wavelet analysis. This method decomposes an image into four bands: HH, HL, LH, LL, then reduces the resolution of the image [4], [5]. In this method, characteristics observed in each band are analyzed and the overall characteristics are detected as a result. Another method is to analyze the block level characteristics after subdividing the image using binary or quad trees [6]. We use the latter method in order to detect details of image structure progressively.

Hereafter, we assume that all images are square, with 2^n pixels per side, and that logarithms are base 2.

A. Hierarchical sorting

As mentioned above, digitally similar density patterns rarely appear in an image. Further, compared to ordinary 8-bit/pixel photographs, 12-bit/pixel medical images have a smaller likelihood for this condition. This tendency explains why methods based on exact pattern matching don't work well for medical images.

Here we propose another fundamental approach to compressing digital images: We pay attention to the order of pixel values and not to the pixel values themselves, because bit depth does not affect the order of the pixel values.

Fig. 1 shows the relationship among the original block, the permutation table and the sorted block. It is clear that the sorted block generates a small predicting code. Our concept, based on hierarchical sorting, is to improve the image compression ratio by choosing the size of a table and using partial sorting. We developed a table generation method that takes into account only the order of the pixels in a block.

A-1. Block division

In a process similar to so-called quad-tree division, the image is divided into four non-overlapping square blocks. The image to be compressed is represented as a square array of pixels. For an image of size $S \times S$, division level L_n ($n = 0,1,2,\cdots,\log S$) is defined. Then, sub-block size $S_B^{L_i}$ at L_i and number of blocks $N_B^{L_i}$ at L_i are defined. A predictive coding method is applied to each block and it produces a code of size $c_p(n)$ for division level L_n. At the same time, the block is sorted according to a permutation table created from the pattern of the block. Then the same predictive coder is applied to the sorted results. The size of code generated by this procedure is $c_s(n)$. The difference of the code sizes, c_l, is computed by

$$C_l = \sum_{n=1}^{S_B^{L_i}} C_p(n) - \sum_{n=1}^{S_B^{L_i}} C_s(n) \cdot \tag{1}$$

We select the coding method c_a for a block based on the value of c_l. If c_l is greater than 0, the permutation table is applied before predictive coding; otherwise, the predictive coder codes the block directly.

All blocks of L_i are tested using (1), then each block is tested with (2) or (3) for whether to do further sub-division.

By dividing an image into small sub-blocks, detailed characteristics of an image can be well understood; however, subdivision of such fine granularity will increase the number of blocks to be processed. In the first step, the entropy of four L_0 blocks and L_1 blocks is compared to that of the code for L_1 blocks that consist of four large blocks.

The necessity of L_1 subdivision is determined by (2).

$$\log \sum_{m=1}^{S_B^{L_0}} c_a(m, L_i) \geq \sum_{n=1}^{N_B^{L_1}} \log \sum_{m=1}^{S_B^{L_1}} c_a(m, n, L_1) \cdot \tag{2}$$

At L_1 subdivision level or higher, it is not necessary to have the entropy of all the sub-images. Estimating the entropy of the entire image at all division levels might overlook the details of the image. Therefore, further subdivision can be performed locally by taking into account only the image area to be processed.

$$\log \sum_{m=1}^{S_B^{L_i}} c_a(m, L_i) \geq \sum_{n=1}^{4} \log \sum_{m=1}^{S_B^{L_{i+1}}} c_a(m, n, L_{i+1}) \cdot \tag{3}$$

The block division process ends when $S / 2^{Li}$ becomes S_T, where S_T is the size permutation table.

A-2. Permutation table creation

We must determine the size of the permutation table before starting hierarchical sorting. Let us define the necessary bits for storing the permutation table as T_{bit}. When the size of permutation table is S_T, T_{bit} is given by $T_{bit} = S_T \log S_T$. As S_T increases, T_{bit} increases rapidly. There are S_T ! different patterns in the set of permutation tables of size S_T. If we make S_T small, one pattern can be applied to many places in the image. However, the result of setting parameters this way is that complex regions of the image get divided into many blocks, so many bits are required to record the inter-block connections. Conversely, if S_T is large, the compression ratio of the region becomes high. However, the necessary storage for the permutation table increases and the likelihood of applying the generated permutation table to other regions decreases. In order to increase the probability of reusing the permutation tables, we use fixed-size permutation tables for all division levels. Hereafter, the word *"resolution"* indicates the number of segmented regions within a block that are formed by quad-tree subdivision or permutation tables. For example, the resolution of a quad-tree block division is 4. If

the resolution of the permutation table is higher than that of quad-tree block division, the sorting of L_i overlaps that of L_{i+1}.

In the complex regions of an image, although rough sorting is done at L_i , more precise sorting is done on L_{i+1}. We call this technique "*hierarchical sorting*." When the size of the permutation table is 2×2, both the resolution of this permutation table and that of the quad-tree block division are 4. In this case, sorting is repeated within each block, and the size of the predicting code may become large in the complex regions of the image. Therefore, a permutation table of size 2×2 is not used with this technique.

As described above, if the size of the permutation table is 8×8 or greater, multiple appearances of the permutation pattern rarely occurs. Hence, permutation tables of size 4×4 are better for our hierarchical sorting.

B. Coding

In hierarchical sorting, an image is divided into two different types of blocks. The block type determines the encoding method. For one type, only predictive coding is applied. This is called "*coding for smooth regions*." For the other type, predictive coding is applied after the block is sorted using permutation tables. This is called "*coding for complex regions*." With one additional bit used to identify the block type, the total bit count of $type_{bit}$ for an image becomes:

$$type_{bit} = \sum_{i=1}^{\log \frac{N}{\sqrt{S_T}}} N_B^{L_i} \, . \tag{5}$$

Each encoding method is explained in the following sections.

B-1. Coding for smooth region

Predictive coding must be done at each division level for progressive decoding. When predictive coding starts at the division level of Li, the average densities of pixels in each block at Li are written to a compressed data stream. Let us denote the number of blocks coded by the coding method at L_j as $NP_B^{L_j}$, the block at L_j as $B_m^{L_j}$ ($m = 1,2,3\cdots, NP_B^{L_j}$), and the n^{th} pixel value in $B_m^{L_j}$ as $v_{B_m^{L_j}}(n)$. The $v_{B_m^{L_j}}$ is the average density value of $B_m^{L_j}$, so

$$v_{B_m^{Lj}} = \frac{\sum_{n=1}^{S_B^{L_j}} v_{B_m^{Lj}}(n)}{S_B^{L_j}} . \tag{6}$$

The predictive coder encodes all $v_{B_m^{L_j}}$ generated by (6). Then the entropy coding is applied to the result. At further division level, the difference of $v_{B_m^{L_j}}$ and $v_{B_m^{L_{j+1}}}$ is encoded by the predictive coder until the size of divided block equals that of permutation table. The predictive coder encodes all differences of the blocks at the division level and the entropy coding is applied to the generated code set. The results are written in the low order of division level.

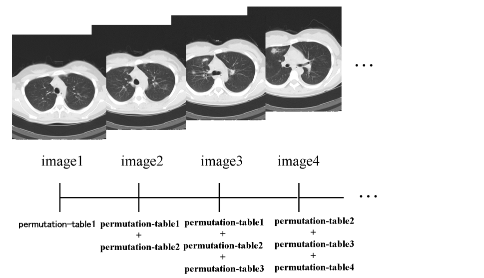

Fig.2. Image set compression: sharing permutation-tables among images

B-2. Coding for complex region

The permutation tables at all division levels sort all blocks, and the predictive coder encodes the pixels in these blocks. The result is sent to the entropy coder. Let us denote the first division level of the block-applied permutation table as L_k . If L_k is higher than L_j, we can use the average of the previous division level or L_{k-1}. If the block at L_k is complex and the blocks at L_{k+1} are smooth, this method writes the average of the pixels in the L_k block.

If the block at L_k and higher blocks use permutation tables, the pixels in the block are encoded by using all of the permutation tables generated at the L_k and higher division levels. In this case, the progressive structure breaks down. However, the length that sequential decoding requires is only the block size at L_k.

Finally, all generated codes are combined. The compressed data consists of permutation tables, the table of block type, and entropy codes of the smooth and complex regions.

III. Image set compression

The permutation tables used by our method are of fixed size. If the image set has similar characteristics, the same permutation tables can be applied to many regions of many other images.

We proposed an image set compression method based on this assumption. This approach is based on hierarchical sorting. Fig. 2 illustrates the process in which the permutation tables of the previous image in the set are applied to the current image. In Fig. 2, the current image references only permutation tables in the previous two images, in order to reduce the number of code bits used for referencing.

IV. Implementaion

An image compression system using hierarchical sorting is independent of the pixel scan and of the predictive coding methods. In our implementation, we used the Hilbert curve to scan an image. We used simple DPCM as the predictive coding method. We showed in Fig. 3 the blocks to which the permutation tables were applied. In our case, we used 4×4

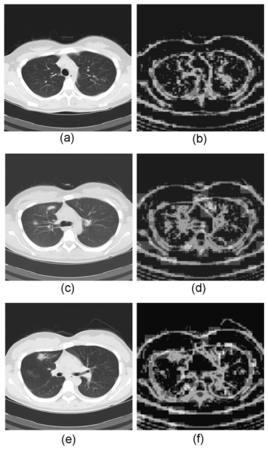

Fig. 3. Examples of blocks in which hierarchical sorting is applied as the image set compression method

pixel permutation tables, with no limit on the number of permutation tables. Fig. 3 shows an example of the blocks to which the hierarchical sorting method was applied for X-ray CT images.

In the example, we used a 512×512 medical image in 12-bit/pixel format. Fig. 3 is an example of our image set compression picked out from the image sequence in the set of sample images. Fig. 3 (a)(c)(e) were original image; (b)(d)(f) are permutation tables generated using the image set compression method. In the complex regions of the image, permutation tables were used at several division levels. These regions were highlighted in (b)(d)(f). It is obvious that many more permutation tables were used for images (c) and (e) than for image (a). Our image set compression used the permutation tables that were generated for previous images.

V. Results and discussion

Table 1 shows details of sample images that were used to compare performance to that of the standard JPEG-LS method. Fig.4 and Fig.5 show performance comparison of the proposed and standard JPEG-LS compression methods. As shown in Fig. 4, when the proposed method was applied to a single image, our method showed poorer performance than JPEG-LS for smaller-sized images (256×256). However, it showed superior performance when the image resolution was greater than 512 × 512. Fig.5 shows the image set compression ratios, normalized by the compression ratios of JPEG-LS. As seen in Fig.5, our method outperformed conventional methods for all image sizes.

Table 1. Sample images

Pixel size	No. of images	Category
256x256	100	MRI images
512x512	20	X-ray helical CT images (chest)
1024x1024	10	gastric barium DR images
2048x2048	10	CR (chest) images

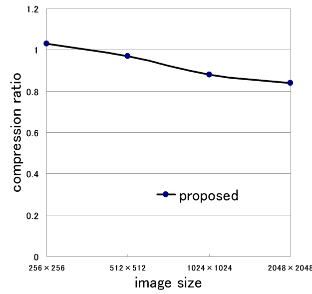

Fig. 4. Compression ratio for medical images normalized by compression ratio by JPEG-LS (single image compression)

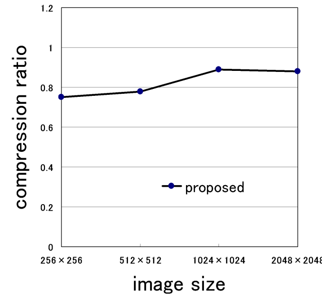

Fig. 5. Compression ratio for medical images normalized by compression ratio by JPEG-LS (image set compression)

VI. Conclusion

In this paper, we proposed a lossless image compression method for medical images, based on hierarchical sorting. Our method takes into account both the global properties of the image and the local complexity of the pixels. We implemented the method and compared performance to a conventional JPEG-LS compression scheme. The results show that our method achieved a compression ratio the same as or better than that of JPEG-LS when applied to a stand-alone image. Furthermore, as our method is designed to handle image sets produced by medical instruments, we confirmed that it outperforms conventional methods when applied to large sets of medical images.

Acknowledgments

The authors wish to thank Dr. T. Kodama for providing medical images.

References

[1] J.B. O'Neal Jr., "Differential pulse code modulation with entropy coding," *IEEE Trans. Inform, Theory*, Vol. IT-21, pp.169-174, March 1976.

[2] ISO/IEC 10918-1, "Digital compression and coding of continuous-tone still images: Requirements and guidelines," February 1994.

[3] M. Weinberger, G. Seroussi, G. Sapiro, "LOCO-I: A Low-Complexity, Context-Based, Lossless Image Compression Algorithm," *Proc. IEEE Data Compression Conference, Snowbird, Utah,* March-April 1996.

[4] L. Wang and M. Goldberg, "Progressive image transmission using vector quantization on images in pyramid form," *IEEE Trans. Commun.*, Vol. COM-37, No.12, pp.1339-1349, December 1989.

[5] Amir Said and William A. Pearlman, "An image multi-resolution representation for lossless and lossy compression," *IEEE Trans. Image Processing*, Vol.5, No.9, pp.1303-1310, September 1996.

[6] N. Ranganathan, Steve G. Romaniuk and Kameswara Rao Namuduri, "A Lossless Image Compression Algorithm Using Variable Block Size Segmentation," *IEEE Trans. Image Processing*, Vol.4, No.10, pp.1396-1406, October 1995.